Transportation Research Part C 101 (2019) 18–34

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

# DeepPF: A deep learning based architecture for metro passenger flow prediction

Yang Liu, Zhiyuan Liu*, Ruo Jia

*Jiangsu Key Laboratory of Urban ITS, Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies, School of Transportation, Southeast University, China*

ARTICLE INFO

*Keywords:*
Passenger flow prediction
Deep learning architecture
Domain knowledge

ABSTRACT

This study aims to combine the modeling skills of deep learning and the domain knowledge in transportation into prediction of metro passenger flow. We present an end-to-end deep learning architecture, termed as Deep Passenger Flow (DeepPF), to forecast the metro inbound/outbound passenger flow. The architecture of the model is highly flexible and extendable; thus, enabling the integration and modeling of external environmental factors, temporal dependencies, spatial characteristics, and metro operational properties in short-term metro passenger flow prediction. Furthermore, the proposed framework achieves a high prediction accuracy due to the ease of integrating multi-source data. Numerical experiments demonstrate that the proposed DeepPF model can be extended to general conditions to fit the diverse constraints that exist in the transportation domain.

## 1. Introduction

As an important public transit system component, the metro system has received increasing attention both academically and practically due to its merits of large capacity, high speed, and high reliability. Thus, the urban rail system has been developing fast while the rapid growth has caused a series of problems, such as over-saturation in/at the train, platform, and other subway facilities (Zhong et al., 2016). To solve these problems, robust prediction of short-term metro passenger flow is of prime importance. It can help the metro authorities to manage the demand and optimize the timetable (Niu and Zhou, 2013; Wu et al., 2015) while enabling passengers to make informed travel decisions (destination choice, departure time, and routing decisions).

Many machine-learning approaches (MLA) like linear regression, feedforward neural network, etc. have been proposed to improve the prediction accuracy of short-term traffic/passenger flow (Hansen et al., 1999; Pereira et al., 2015). Additionally, various deep learning methods have also been used in the literature (Lv et al., 2014; Ma et al., 2015; Zhang et al., 2017; Gu et al. 2018; Tang et al., 2018). Although MLA has become an emerging and substantial technology in the transportation area (Miles and Walker, 2006; Liu et al., 2018); the challenge to efficiently use it for metro passenger flow prediction persists due to the complex nature of transport systems and is therefore taken as the focus of this paper.

The flow prediction task can be regarded as a time series prediction problem as it varies with time. Thus, analysis of the temporal dependencies in the variable of interest is of prime importance. In this regard, the existing methods (both conventional and emerging deep learning methods) generally use the data from a recent time interval to make the prediction. However, two main characteristics of the time series i.e. cycle and trend are not fully considered in past studies. Herein, the cycle reflects a fixed cyclical pattern when

* Corresponding author.
*E-mail address:* zhiyuanl@seu.edu.cn (Z. Liu).

https://doi.org/10.1016/j.trc.2019.01.027
Received 16 May 2018; Received in revised form 28 January 2019; Accepted 28 January 2019
Available online 13 February 2019
0968-090X/ 

time series is not subjected to any extreme influencing factors while trend reflects the long-term variation of the time series (e.g. a rising trend). The metro passenger data is generally associated with temporal characteristics that are repetitive at fixed time interval e.g. the metro passenger flow at a certain time interval may be similar to that of the same time interval on the previous day; depicting that 24 h should be taken as the cycle period.

Proper integration of the above two characteristics into the prediction models could considerably improve their performance in real-world applications. Nevertheless, for different transportation problems, these characteristics have different implications. Further, it should be noted that no standard routine/method exists as a golden rule to solve the diverse transportation flow prediction problems. Hence, it is essential to investigate the nature of these characteristics in metro flow prediction.

Furthermore, the metro system has numerous unique operational properties (e.g., unlike buses, the trains follow a more precise timetable) that should also be integrated while designing MLA for metro passenger flow prediction. Such endogenous characteristics/properties of a transportation system (also known as domain knowledge) should be considered for the rational design of MLA. Therefore, this paper aims to develop an end-to-end deep learning-based architecture for the short-term metro passenger flow prediction that incorporates the domain knowledge.

### 1.1. Literature review

As a seminal work, Ahmed and Cook (1979) investigated the application of the Autoregressive Integrated Moving Average (ARIMA) model for short-term prediction of freeway traffic flow. Regressive time series models like ARIMA model and its variants are widely used for prediction tasks, for example, the multivariate ARIMA (ARIMAX) model was applied to motorway data and found to outperform the univariate forecast models (Williams, 2001). Williams et al. (1998, 2003) proposed a theoretical basis for modeling traffic data as seasonal ARIMA (SARIMA) processes. Apart from the variants of ARIMA models, Kalman filtering models are also popularly used as its prediction accuracy is unaffected by noise in the data. Jiao et al. (2016) revised the traditional Kalman filtering model based on the error correction coefficient, historical deviation, and Bayesian combination to predict short-term rail transit passenger flow.

However, although the changes in traffic data are nonlinear in nature (mainly due to the frequent switching of traffic conditions from free flow to congested status and vice-versa), all the above-mentioned models are limited in their performance and application due to the assumption of linearity (Zhang et al., 1998; Polson et al., 2017). Therefore, various models/methods like machine learning models (e.g., Support Vector Machine) and deep learning models (e.g., Convolutional Neural Network) that overcome such limitations have been developed and applied in traffic or passenger flow prediction.

Various machine learning-based methods have been developed to handle the nonlinear features of traffic data efficiently. Sun et al. (2006) proposed a traffic flow prediction model for both complete and incomplete data based on the Bayesian network. To deal with the frequent fluctuations and abrupt changes, Chang et al. (2012) proposed a dynamic multi-interval method based on the k-nearest neighbor regression for traffic flow prediction. Jeong et al. (2013) proposed a novel online learning weighted support-vector regression prediction model for short-term traffic flow forecasting.

Furthermore, various neural network models have also been developed for traffic/passenger flow prediction. Conventional feedforward neural network (e.g., Back Propagation neural network) can capture the complex nonlinear relationship without prerequisite knowledge (Zhang et al., 1998). Vlahogianni et al. (2005) proposed a multilayered structural optimization strategy based on genetic algorithm to better represent the traffic data with spatio-temporal characteristics. Chan et al. (2012) employed the Levenberg-Marquardt method and the hybrid exponential smoothing approach to train the neural network and reported that the algorithm significantly improved the generalization capabilities for short-term traffic flow prediction.

Recently, deep learning has attracted considerable academic and industrial interests (Bengio et al., 2009). It has been successfully applied in different domains like natural language processing, computer vision, speech recognition, etc. (Collobert et al., 2008; Hinton et al., 2012, Krizhevsky et al., 2012). In past studies, many deep learning methods have been applied for traffic flow prediction as they can capture the complex nonlinear relationship and the latent correlation features in traffic flow data. Wei et al. (2012) proposed a hybrid model by combining empirical mode decomposition method and neural network for short-term passenger flow prediction in subway systems. Sun et al. (2012) investigated the network-scale modeling approach to short-term traffic flow prediction. They proposed a multilink, single-task model that combined graphical lasso with neural network and concluded that it improved the prediction accuracy for network-scale traffic flow forecasting. Lv et al. (2013) presented a stacked autoencoder (SAE) neural network for traffic volume prediction to capture the general traffic flow feature representation from the raw data. Gang et al. (2015) adopted SAE to predict continuous travel time for transit signal priority. The SAE model was also modified to impute the missing traffic data and to suit different research periods (Duan et al. 2016a, 2016b). Ma et al. (2015) used the recurrent neural network (RNN) for traffic speed prediction while Duan et al. (2016c) used its variant, LSTM for travel time prediction. Zhang et al. (2017) partitioned a city into a grid map and presented a novel spatio-temporal residual network to forecast the crowd flows in each region. The use of convolution neural networks captures the spatial dependencies in data. Huang et al. (2014) proposed a deep architecture that consists of a deep belief network (DBN) and a multitask regression layer to predict short-term traffic flow. The positive results demonstrated that deep learning is a sound tool for traffic/passenger flow prediction. Traffic flows have sharp nonlinear features due to the frequent changes in free flow, congestion, breakdown, and resumed situations. To tackle these challenges, Polson and Sokolov (2017) proposed a deep learning architecture to capture the nonlinear spatio-temporal effects.

Various issues of public transportation like bus scheduling (Bie et al., 2015), bus stop-skipping (Liu et al., 2013), boarding/alighting behaviors (Liu et al., 2016), metro timetable design (Niu and Zhou, 2013; Wu et al., 2015), etc. have been addressed in past literature. Many theoretical achievements have also been successfully implemented in practice as well. Therefore, when designing the

novel/emerging methods like MLA, the findings and achievements of the previous studies should be fully considered.

Some gaps in the existing studies on metro passenger flow prediction are observed and summarized as follows: (i) only the flow data from recent time intervals are used to make the prediction while the unique properties and domain knowledge of the metro system (e.g., the timetable) are often neglected. (ii) The influential factors and the fundamental reasons for the accuracy in forecasting results should be further analyzed rather than just following the beaten track from other domains. (iii) As the studies of big data in different disciplines (e.g., computer science, transportation, and communication) are relatively inter-dependent; inter-disciplinary coordination is needed. These gaps are thus addressed in this paper. Moreover, with the challenges associated with growing metro demand, it is a timely topic to address the MLA for metro passenger flow prediction.

### *1.2. Objectives and contr ibutions*

This paper aims to solve the metro passenger flow prediction problem by integrating the modeling skills of deep learning and the domain knowledge in transportation. Apart from using the flow data in a recent time interval, we further consider two temporal properties. The fundamental causes of the achieved forecasting accuracy were holistically analyzed to promote its application in various general scenarios.

Most studies incorporate spatial characteristics by dividing the city into grids according to geographical location and then capture the implicit spatial relationship between areas. It is based on the idea that the inflow of a particular zone is affected by outflows of nearby regions as well as distant regions. Nevertheless, the metro network has its own unique topological structure. As the geographically adjacent stations may not be connected, the spatial characteristics cannot be used directly as in the case of the road network. Thus, the correlation between stations should not be measured by their geographical distance. Considering the unique topological structure of the metro network, we use the average travel time instead of the geographical distance to represent the spatial characteristics. We further construct features through inflow and outflow conservation to capture the spatial characteristics between stations. Inspired by the findings in urban rail schedule (Niu and Zhou, 2013; Wu et al., 2015), we proposed a completely data-driven method by taking into consideration the train arrival time, passengers' walking time, and the lag time caused by congestion.

The contributions of this paper are four-fold. First, a deep-learning architecture based on the spatio-temporal characteristics of metro data is proposed. Herein, the architecture has multiple extensible components, including external environmental factors, temporal dependencies, spatial characteristics, and metro operational properties. Second, the temporal properties of passenger flow are revealed (i.e. the passenger flow in recent time, daily periodicity, and weekly trend). Third, to capture the implicit spatial relationship between stations, we propose a passenger flow conservation strategy to forecast the sudden increase or decrease in passenger flow. Fourth, the metro operation has distinctive characteristics (e.g., the trains run on more precise timetables). The forecasting accuracy of outbound passenger flow during a short time period is restricted by the timetables. Hence, we proposed a completely data-driven method taking the arrival time, passengers' walking time, and the lag time caused by congestion into consideration. The latest timetable can be learned from the raw passenger flow data.

The remainder of this paper is organized as follows; Section 2 defines the context of the addressed problem. Modeling techniques are described in Section 3. Section 4 presents a deep-learning-based metro passenger flow architecture. Section 5 presents the experiments that are carried out to verify the proposed deep learning approach along with the in-depth analysis of the results. Finally, Section 6 concludes this paper.

## 2. Problem statement

The data used in this study are standard metro service/transaction data that is collected each time a passenger uses the IC card to enter or leave a metro station. Similar data tags are recorded in the central computer system of the city's metro system, including six tags namely; user id, inbound station, outbound station, inbound time, outbound time, and type of card. It should be noted that the personal information in the dataset is anonymized to protect the passengers' privacy.

Within a fixed interval (e.g., 10 min interval), the total inbound (enter the station) and outbound (exit the station) passenger flow could be easily calculated based on the metro service data. We use $x_t$ to denote the outbound or inbound passenger flow in the $t$-th time interval. The passenger flow prediction is evidently a time series forecasting problem as values of consecutive intervals show temporal dependence. Thus, the problem addressed in this paper is to use the historical passenger flow data $\{x_{t-1}, x_{t-2}, x_{t-3}, ...\}$ to predict $x_t$.

With these data, one major limitation in the forecasting of passenger flow is that the existing analysis is carried out under ideal conditions, without considering various complex scenarios that prevail in real conditions. It's very challenging to accurately estimate the passenger flows because the raw data is collected in different scenarios. Such a scenario consists of a combination of the following cases:

1. There is no detailed train schedule data. The train schedule refers to the real-time operational data with station id and specific train arrival time.
2. We only possess data from a single station. The data of the whole train line or the entire metro network is missing.
3. Only the historical passenger flow observations in recent time intervals are available for use.

This paper thus aims to build a purely data-driven MLA based on the deep learning technique to overcome the above-mentioned

challenges in metro flow prediction. In the proposed deep-learning-based architecture, we consider multiple extensible components to model external environmental factors, temporal dependencies, spatial characteristics, and metro operational properties in short-term metro passenger flow prediction.

## 3. Prerequisite of deep learning models

To address the multiple extensible components mentioned above, we propose a novel deep-learning-based architecture. The feedforward neural network (e.g., back- propagation neural network, BPNN) is a straightforward and commonly adopted MLA. In this section, the limitation of BPNN in solving the addressed issue is first elaborated. Then, a recurrent neural network (RNN) and its variant, long short-term memory neural network (LSTM) is introduced to build the deep-learning-based architecture. In contrast with BPNN, the rationale and superiority of the proposed architecture are summarized.

### 3.1. Feedforward neural network

The BPNN with a single hidden layer is one of the simplest feedforward neural networks where BP refers that the training of neural networks is carried out by using backpropagation. Moreover, BP can be used in both deep and shallow models. Various other approaches (e.g., ELM) are also used to train the feedforward neural networks (Huang et al.,2006). Taking the values in previous intervals $\{x_{t-1}, x_{t-2}, x_{t-3}, ...\}$ as a vector, the BPNN then predicts $x_t$. It is a supervised learning problem with the label, $x_t$. The forward propagation process of feedforward neural network can be expressed as follows:

$$z^{(l+1)} = W^{(l)}a^{(l)} + b^{(l)} \tag{1}$$

$$a^{(l+1)} = f_l(z^{(l+1)}) \tag{2}$$

where $l$ denote the $l$-th layer in the neural network, $W^{(l)}$ denote the weight associated with the connection between layer $l$ and layer $l+1$. $a^{(l)}$ is the activation value in layer $l$, and $a^{(1)}$ is the values from the input layers. $b^{(l)}$ is the bias, $z^{(l)}$ denote the total weighted sum of inputs, and $f_l(\cdot)$ is the activation function. Considering a case where the window size, $n = 3$ and $l = 1$;

$$a^{(2)} = f_1(W^{(1)}concat(x_{t-1}, x_{t-2}, x_{t-3}) + b^{(1)}) \tag{3}$$

where $concat(\cdot)$ represents splicing $x_{t-1}$, $x_{t-2}$, $x_{t-3}$ into a vector of a larger dimension.

From Eq. (3), we can see that the feedforward neural network uses the data from past three time intervals $x_{t-1}$, $x_{t-2}$, $x_{t-3}$ to train the network once, i.e., cannot use data efficiently. The initial purpose of developing deep learning technologies was to solve problems associated with traditional information technology domain like computer vision rather than solving traffic problems. Thus, the architecture of the existing models needs to be modified to enhance their suitability in traffic studies. As the time span of the time series data is limited, it is impossible to construct a large number of samples. Moreover, BPNN adopts the form of a fully connected layer rather than parameter sharing; resulting in too many parameters and requiring more data to train the network. Thus, these aspects make the feedforward neural network model unsuitable for solving time series problems in transportation. However, it should be noted that the feedforward neural network performs well in capturing the nonlinear relationship of the data, especially when useful features are already constructed in the feature engineering stage. Therefore, while designing the structure of neural networks, we adopt a recurrent neural network for time series data while the feedforward neural network is used for the non-time series data.

### 3.2. Recurrent neural network and the long short-term memory

In the feedforward neural network, we assume that all inputs are independent of each other. However, time series data are intrinsically continuous rather than discrete. Hence, time series seldom produce sharp variations and the passenger flow at a certain time interval has a close resemblance to flow at the adjacent time segment. The term recurrent in RNN implies that the model's output depends not only on the current computations but also on the previous computations. In RNN, the activations conducted at the hidden layer are from both the current input and the hidden layer activations from previous timesteps (Gers et al., 2000; Graves et al., 2013).

Fig. 1 shows a simple RNN architecture with a single input unit and a single recurrent hidden unit. $x_t$ is the input at time step $t$, $S_t$ is the hidden state at time step $t$ and is also the 'memory' of the network. We can see that the current state $S_t$ is not only dependent on the current input $x_t$ but also on the previous state, $S_{t-1}$. $S_t$ is calculated in terms of the previous hidden state and the input at the current step; given as follows:

$$S_t = f(Ux_t + WS_{t-1} + b) \tag{4}$$

where the function $f(\cdot)$ is usually a nonlinear function. $U$ and $W$ are the weight matrixes, which are left unchanged for all the layer. Based on the Eq. (4), the hidden state at time step $t-1$ and $t-2$ is obtained as follows;

$S_{t-1} = f(Ux_{t-1} + WS_{t-2} + b)$, and $S_{t-2} = f(Ux_{t-2} + W \cdot S_{t-3} + b)$

While training the model with the values of $U$ and $W$, RNN can use values of the previous time intervals $x_{t-1}$, $x_{t-2}$, $x_{t-3}$ for three times as it allows parameter sharing whereas BPNN can only use these values once in the case mentioned in Section 3.1. This advantage enables RNN to account for the contextual dependency in the input data. However, the performance of RNN is limited in many cases, mainly due to the vanishing gradient problem that limits the length of the input (Hochreiter, 1997). To overcome this

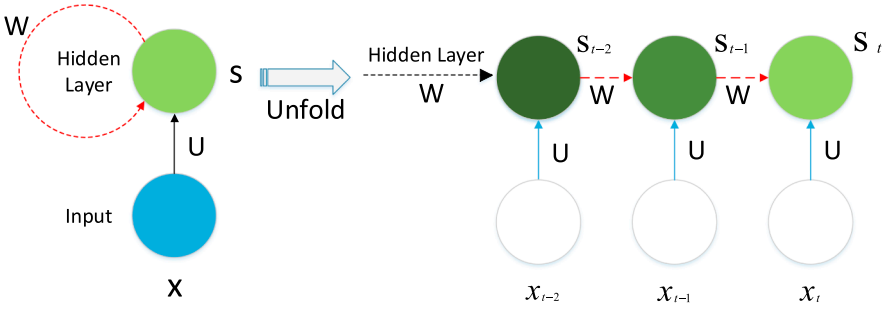


**Fig. 1.** Illustration of a single input unit and a single recurrent hidden unit.

hurdle, Hochreiter and Schmidhuber (1997) developed the long short-term memory (LSTM) method. LSTM is explicitly designed to avoid the long-term dependency problem.

As illustrated in Fig. 2, the architecture of LSTM consists of memory cells as basic units. The memory block in LSTM consists of a cell, input gate, forget gate, and the output gate. The gates can optionally retain the information. These simple changes enable the cell to store and read long-term contextual information, thus, solving the vanishing gradient problem.

Input gate:

$$i_t = \text{sigmoid}(W_i[h_{t-1}, x_t] + b_i) \tag{5}$$

Forget gate:

$$f_t = \text{sigmoid}(W_f[h_{t-1}, x_t] + b_f) \tag{6}$$

Output gate:

$$o_t = \text{sigmoid}(W_o[h_{t-1}, x_t] + b_o) \tag{7}$$

where $x_t$ is the input at time $t$, $h_{t-1}$ is the hidden state at time $t-1$. $W_i$, $W_f$, and $W_o$ are the weight matrices of the input gate, the forget gate and the output gate respectively while $b_i$, $b_f$, $b_o$ are their respective bias vectors.

Given an input $x_t$ at time $t$, cell state $c_{t-1}$ and hidden output $h_{t-1}$ at time $t-1$, the cell state $c_t$ and hidden output $h_t$ can be obtained as follows:

Cell state:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c[h_{t-1}, x_t] + b_c) \tag{8}$$

Hidden output:

$$h_t = o_t \odot \tanh(c_t) \tag{9}$$

The operator '$\odot$' is a Hadamard product. tanh and sigmoid are two commonly used nonlinear activation functions; given as

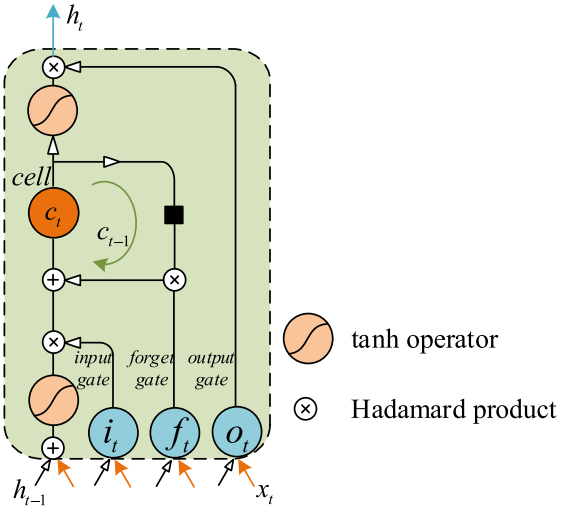


**Fig. 2.** Illustration of the hidden layer unit of LSTM model.

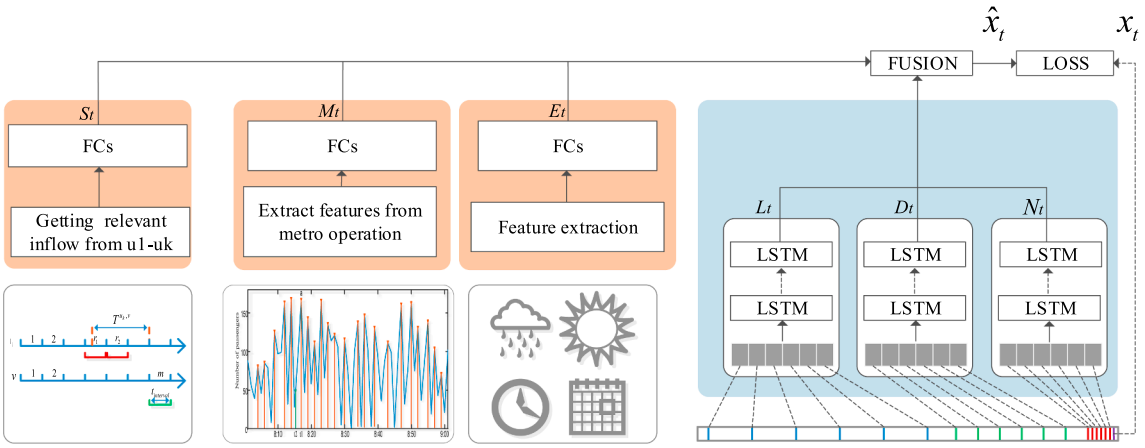


**Fig. 3.** Deep learning-based passenger flow architecture.

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{10}$$

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \tag{11}$$

## 4. Deep learning-based metro passenger flow architecture

Based on the LSTM, we propose an end-to-end deep learning-based architecture (shown in Fig. 3) which can reasonably address the input features of the metro passenger flow prediction problem. Many traditional machine learning systems are composed of multiple individual modules. The training of these modules is often performed individually, with separate objective functions. The objective functions might not always be consistent with overall performance indicator of the system. Thus, the resulting performance of the trained network is not optimized. The purpose of end-to-end is integrating individual modules, which is also the focus of the model in this paper. The architecture comprises multiple extensible components, including modeling external environmental factors, temporal dependencies, spatial characteristics, and metro operational properties respectively.

It should be noted that the neural network architecture could be layer-specific i.e. the different layer can take different neural network model, for instance, a mixture of LSTM and convolutional neural network. Herein, we use a modular design, which defines an LSTM module with multiple LSTM layers to temporal dependencies in data. When designing a deep learning architecture with multi-source input data, an individual module containing multiple layers of the network is designed for each type of data while the modules are executed in parallel. For every module, the hidden layers are in sequential order.

### 4.1. External factors

Passenger flow can be affected by various external factors such as weather and holidays. To analyze the effects of these factors, the passenger flows under different circumstances are thoroughly compared. The precipitation data obtained for the study records the grade of precipitation. As shown in Fig. 4(a) and (b), compared to a normal day, holiday and weekend have an evident influence on the passenger flows. For instance, if heavy rain is observed on Monday, we will then compare it with the data of previous Monday to ensure that they have the same characteristics. While considering the flow of the entire metro network, weather (heavy rain) is found to have a negligible impact, as observed in Fig. 4(c). However, on a few stations, it is found to sharply reduce the passenger flows compared to the same day of the latter weeks, as shown in Fig. 4(d). It should be noted that there are only a few days with records of heavy rain in the dataset. Fig. 4(c) and (d) shows one of the contrastive observations.

We note that for the analysis of passenger flows, the day of week and time of day features have a significant impact on the accuracy of the prediction and thus, should be considered. For example, many researchers are found to use only flow at previous time intervals i.e.$\{x_{t-1}, x_{t-2}, x_{t-3}, \ldots\}$ to predict $x_t$. Assuming there are two samples $\{x_t^1, x_{t-1}^1, x_{t-2}^1, x_{t-3}^1\}$ and $\{x_t^2, x_{t-1}^2, x_{t-2}^2, x_{t-3}^2\}$, even though their previous traffic volumes are same, there is a big difference between $x_t^1$ and $x_t^2$, e.g., $x_t^1$ is in the peak hour while $x_t^2$ is not. It refers that even though the features are the same, the label has changed and thus, ignoring features like the day of week and time of the day may be the probable reasons for the reduction in prediction accuracy. If the performance of the model was not improved by the inclusion of these features, it does not mean that they are insignificant, but it requires further analysis. Further study is needed to determine the applicable conditions for certain features and how to improve the training to achieve better results.

In an actual application scenario, there are a lot of features that are not continuous numerical variables, but discrete categories. One hot encoding is a common way to encode categorical features. For example, suppose '5' indicates Friday, and for a whole week, its one hot representation shall be (0, 0, 0, 0, 1, 0, 0). In one hot representation, each day is an independent dimension. However, this

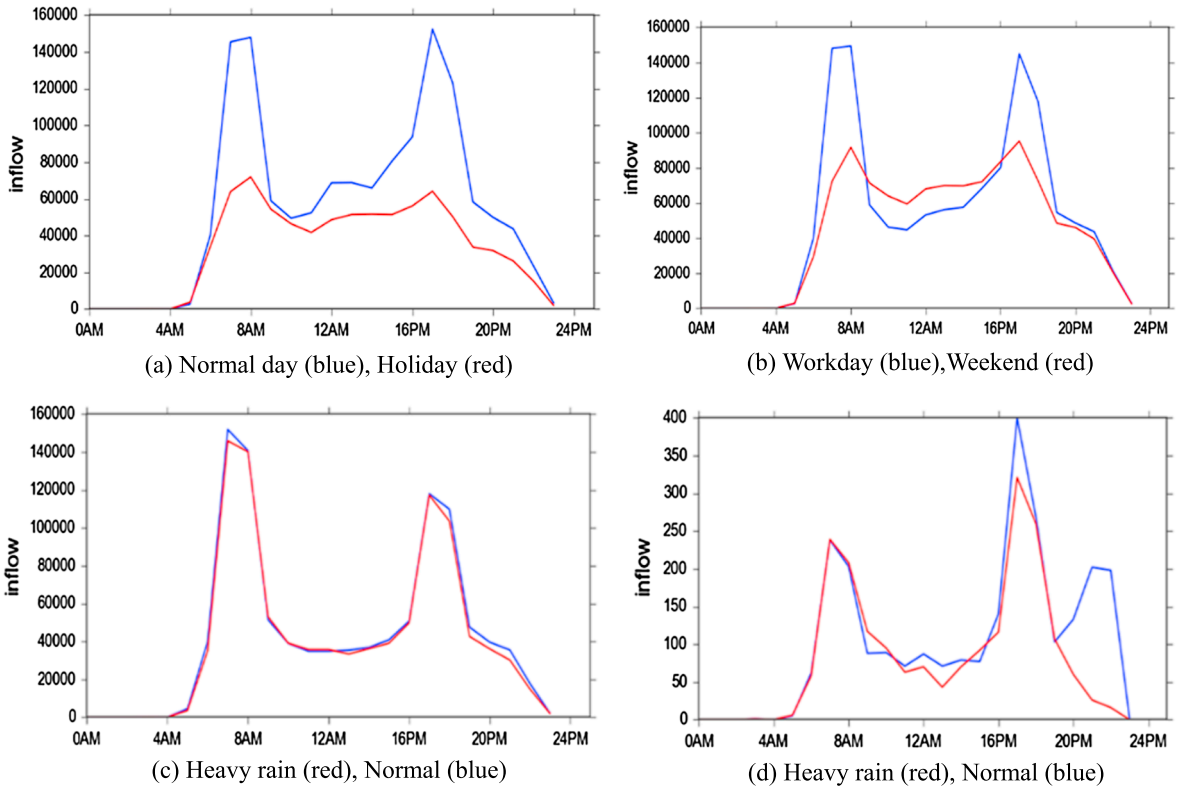


**Fig. 4.** Inbound passenger flow.

characteristic is inconsistent with the actual situation (e.g., workdays may be similar). Moreover, when the number of categories is large, this is too sparse for one hot encoding. Further, the deep learning models does not work well with the highly-sparse dimensional feature.

Embedding technique is used to solve the above-discussed problems associated with one hot encoding. Embedding technique is a popular method widely used in natural language processing (Bengio et al., 2006), which can map categorical values into a dense vector. Embedding can be represented as a mapping:

$$Y = E(X) \tag{12}$$

where the function $E(\bullet)$ is injective and has a structure-preserving characteristic (e.g., $X_1 < X_2$ in the space where $X$ belongs to; when mapping into $Y$ space, likewise, $Y_1 < Y_2$). To handle the categorical features, an embedding layer is built with parameter matrix $W \in R^{I \times O}$, where $I$ denotes the number of categories and $O$ denotes the dimension of the embedding space (i.e., out space). Generally, one hot encoding is used to represent categorical features. For certain category $i \in [I]$, it is represented by $onehot(i) \in R^{1 \times I}$ and is then converted to an embedded vector $embed(i) \in R^{1 \times O}$, which is equivalent to multiplying $onehot(i)$ by parameter matrix $W$ (i.e., the $i$-th row of $W$). In most cases, $O$ is far smaller than $I$. Therefore, the categorical values with high-dimensional one hot representation can be efficiently fed to the model and processed. By embedding method, in the above case, Friday can be represented in the form of (0.3, 0.1). Suppose that the passenger flow on Thursday is similar to Friday; then their representation is also similar (it is learned through neural networks). In our model, we use this technique to find out the similarities between different days and hours.

As shown in Fig. 5, we used three Embedding Layers to embed the features; namely the day of the week, the time of day, and the

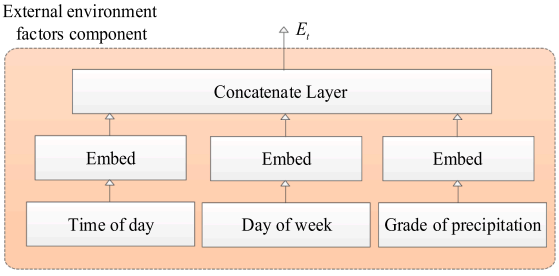


**Fig. 5.** External environmental factors component.

grade of precipitation. A concatenate layer then concatenates their outputs. The concatenated output of this component at $t$-th time interval is denoted as $E_t$.

### 4.2. Temporal dependencies

Clearly, the flow at the current time interval shares a higher similarity with those in recent time intervals than the distant ones. Existing studies often use the flow in the recent time interval to make the prediction. The temporal dependency is represented as $N_t = [x_{t-1}, x_{t-2}, ...,x_{t-c}]$, where $c$ is the number of dependent timestamps. Apart from the flows in recent time intervals, we further consider daily periodicity and weekly trend for the proposed model.

A time series has two key characteristics namely; (1) cycle and (2) trend. The traffic/passenger flow data usually possess an evident periodic pattern, for instance, the metro passenger flow at a particular time interval of a day may be similar with that of the previous day; implying the cycle of 24 h.

Another implication of the cycle can be observed in the case of the commuting passengers in the metro. The metro passenger flow data reflects the passengers' fixed behavior patterns reasonably well. This is one of the reasons for the improved prediction accuracy while considering time cycle and trend characteristics. Similarly, in the case of the road network, the supply and demand of taxis in urban areas and the traffic flow on urban roads have similar temporal characteristics (i.e., cycle and trend). Thus, daily periodicity characteristic is applied to our passenger flow forecasting model.

We take the period segment to depict the above daily periodicity, and it is defined as $D_t = [x_{t-d}, x_{t-2d}, ...,x_{t-nd}]$, where $n$ is the numbers of dependent timestamps in the period parts (i.e. use $n$ days of data) and $d$ is a fixed period (e.g. a period of one day, $x_{t-d}$ means the passenger flow at the same time interval in the previous day).

Here, the trend parts are used to describe the weekly trend, and it is defined as $L_t = [x_{t-w}, x_{t-2w}, ...,x_{t-mw}]$, where $m$ is the numbers of dependent timestamps in the trend parts and $w$ is a fixed trend span (e.g., a trend span of one week, $x_{t-w}$ means the passenger flow at the same time interval in the previous week).

As shown in Fig. 3, for each temporal property (i.e. the passenger flow at adjacent time intervals, daily periodicity, and weekly trend), we stack 3 LSTM layers, enabling the model to learn higher-level temporal representations. In this component, we use $N_t$, $D_t$, and $L_t$ defined above to represent the three temporal properties.

### 4.3. Spatial characteristics

The extraction of spatial characteristics can be divided into two categories, namely; manual feature design and automatic extraction through a convolutional neural network. Manual feature design requires a high level of experience and domain knowledge, and can still be applied to data that lacks a grid-like structure.

The convolutional neural network is suitable for data with grid-like structure, for example, images and audios that can be converted to two-dimensional and one-dimensional grids respectively (Goodfellow et al., 2016). In case of a single metro line or road, it can be regarded as a one-dimensional grid. Thus, we can use a one-dimensional convolutional neural network to automatically extract the spatial feature, without the need for complex feature engineering (Wu et al., 2018). While using a two-dimensional convolutional neural network, the spatial feature extraction is generally carried out by partitioning the network into grids (see, e.g., Zhang et al., 2017) to capture their spatial characteristics. As shown in Fig. 6(a), the inflow of zone $Z_1$ is affected by the outflow of nearby regions (e.g. $Z_2$ and $Z_3$) as well as distant region (e.g. $Z_4$). Then, the inflow and outflow of location-based data (e.g., taxi service data and bicycle-sharing service data) can be forecasted based on the grids. Theoretically, the above-mentioned spatial characteristics can be better captured by Convolution Neural Network (Zhang et al. 2017), as shown in Fig. 6(b).

The above approach (partitioning the network into grids) is not suitable for the metro system. This is because, in the case of the road network traffic (private vehicles, taxis, bikes, etc.), the adjacent grids are geographically connected and thus, the flows have interactions. However, for the metro system, adjacent stations may not be directly connected (two stations are on different metro lines altogether). In this study, we focus on the whole metro network, which cannot be represented as a grid-like structure directly.

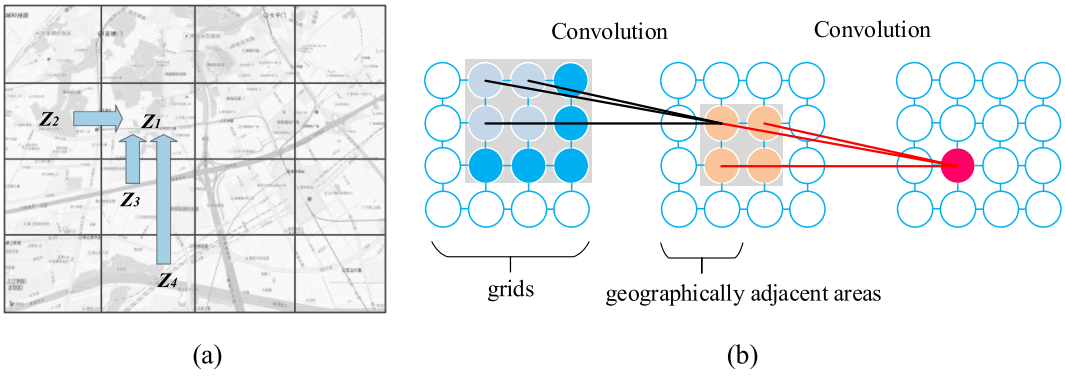


**Fig. 6.** Illustration of the spatial characteristics.

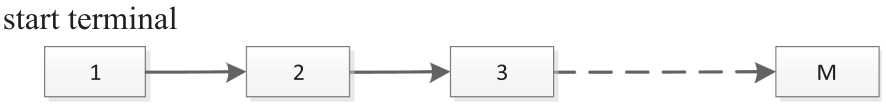


**Fig. 7.** Illustration of a single-directional urban rail line.

Thus, based on our domain knowledge of rail transit, we devised a method that manually designs high-level features to overcome this problem.

Zhang et al. (2017) used the Convolutional Neural Network to capture the implicit spatial relationship between the inflow and the outflow. Therefore, to capture the spatial characteristics of metro stations, new features can be built based on the flow conservation principle that the outflow can always counterbalance the inflow. Herein, the average travel time instead of the geographical distance is taken as a measure of the travel impedance between stations. For the sake of presentation, we use a simple case to elaborate our approach. Assume that the train operates on a single-directional urban rail line with $M$ stations, as shown in Fig. 7. The stations are numbered as 1, 2, …, $M$ sequentially.

Let $I_j^{u,v}$ represent the number of passengers going to station $v$ boarding a train $j$ at station $u$ (inflow) and $O_j^v$ denote the number of passengers alighting from train $j$ at station $v$ (outflow). The inflow and outflow conservation is given as:

$$O_j^v = \sum_{u=1}^{v-1} I_j^{u,v} \; for \; j = 1, 2, ...,K. \tag{13}$$

where $K$ is the total number of trains departing from the start terminal over the operating time period.

$I_j^{u,v}$ at time $t_{in}^u$ will influence $O_j^v$ at time $t_{out}^v$. For example, once $I_j^{u,v}$ increases, $O_j^v$ at the corresponding time $t_{out}^v$ will necessarily increase. Let $T^{u,v}$ denote the travel time between $u$ and $v$:

$$T^{u,v} = t_{out}^v - t_{in}^u \tag{14}$$

The passenger flows are also affected by the crowdedness of the station (undersaturated or oversaturated conditions) (Niu and Zhou, 2013). For undersaturated conditions, all passengers waiting at a platform can board the arriving train. For oversaturated conditions, only the passengers who arrive early can board the arriving train.

However, in practice, it's difficult to obtain the value of $I_j^{u,v}$ in Eq. (13). This is because, based on the IC card data, we can only know a passenger's inbound time to the station but not the exact boarding time to the train. Hence, it's difficult to determine the passengers' in-vehicle travel time, which train they have taken, and the total number of passengers in each train. Note that, for passenger flow prediction, it's inappropriate to use the passengers' destination stations as a feature because it is unknown future information.

In accordance with passenger flow conservation, this paper provides the following method to overcome the inadequacy in IC card data:

**Step 1:** For any destination station $v$, we sort the OD data of a complete day and select only $k$ origin stations $\{u_1, u_2, ...,u_k\}$ with the largest inbound passenger flows, since the metro network is quite large and the inbound passenger flows of some stations are relatively low.

**Step 2:** Calculate the average travel time $\{T^{u_1,v}, T^{u_2,v}, ...,T^{u_k,v}\}$ between station $v$ and $\{u_1, u_2, ...,u_k\}$. Here, the travel time refers to the outbound time minus the inbound time.

The average travel time $T^{u_k,v}$ can be calculated as follows:

$$T^{u_k,v} = \sum_{i=1}^{n} (t_{leave}^{v,i} - t_{enter}^{u_k,i})/n \tag{15}$$

where $t_{enter}^{u_k,i}$ is the time when passenger $i$ enters station $u_k$; $t_{leave}^{v,i}$ is the time when this passenger leaves station $v$. The time here refers to the timestamp in the original data with the precision of minute, which is used to calculate the average travel time. $n$ is the number of passengers travelling from station $u_k$ to $v$. The average travel time can be accurately calculated as IC card data includes the inbound stations, outbound stations, inbound time, and outbound time of all the passengers.

**Step 3:** When forecasting the outbound passenger flow at the $m$-th time interval of station $v$, we need to figure out the inbound passenger flows at which one of the time intervals of station $u_k$ will have a direct impact on it. Based on the average travel time, these time intervals of station $u$ can be obtained as:

$$r_1 = m - \lceil T^{u_k,v}/t_{interval} \rceil \tag{16}$$

$$r_2 = m - \lfloor T^{u_k,v}/t_{interval} \rfloor \tag{17}$$

where $\lceil\rceil$ ($\lfloor\rfloor$) denotes the rounding up (rounding down) operation.

**Step 4:** Acquiring the inbound passenger flow data at the $r_1$-th, $(r_1 - 1)$-th, $(r_1 - 2)$-th,…, $(r_1 - e1)$-th, $r_2$-th, $(r_2 + 1)$-th, $(r_2 + 2)$-th, …, $(r_2 + e_2)$-th intervals of station $u_k$ $(0 \leqslant e_2 \leqslant m - r_2 - 1)$ .

Unlike most of the existing studies that use only passenger flow in recent time to predict, considering inflow and outflow conservation can enhance the predictive capability of the model. Over a certain period of time (i.e. lag time), the fluctuations of inflow at upstream stations will affect the subsequent outflow at the downstream stations. The key aspect of our method is to find the

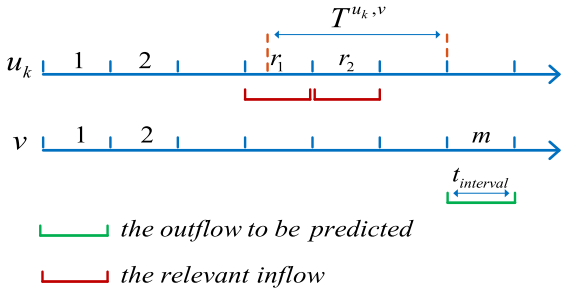


**Fig. 8.** Illustration of the relevant inflow.

relationship between stations by their relative contributions and estimate the lag time by using average travel time. Several fully-connected layers can be composed of a multilayer feedforward neural network, also known as multilayer perceptron (MLP). Feedforward neural network is adopted due to their superior performance on nonlinear problems. Thus, for the non-time series features in this component, we adopt a fully-connected layer. The output of this component at $t$-th time interval is denoted as $S_t$.

### 4.4. Characteristics of metro operation

Metro operation has various distinctive characteristics, for example, compared to buses, the trains run on more precise timetables. Therefore, the forecasting accuracy of outbound passenger flow during a short time interval will be restricted by the timetables. For instance, 3 trains arrive at a station between 8:01–8:10 a.m. while 4 trains arrive between 8:11–8:20 a.m. Fig. 8 shows the outbound passenger flow from 8:00–9:00 a.m. (aggregated at 1-minute interval), with passenger flow peaks depicting the train arrivals. This can be visually interpreted via a heatmap, as shown in Fig. 10. The horizontal axis represents the 60 min in an hour while the vertical axis represents the metro operation time from 7:00 to 20:00. Each grid represents the outbound passenger flow aggregated at the 1-minute interval for a whole day. A grid with deeper color implies large outbound passenger flow (the values have been scaled in the row direction). When a train arrives, passengers will begin to disembark, resulting in a pulse of the outbound passenger flow. As a train usually runs according to the schedule, the pulses will manifest in a regular pattern, which is termed as the *strong impulsive pattern*. A heatmap (Fig. 10) is used to illustrate this pattern. We can observe that the distribution of this outbound passenger flow shows a strong impulsive pattern.

Data visualization helps in illustrating the temporal pattern of outbound passenger flow and demonstrate the relationship between outbound passenger flow patterns in detail. It is necessary to fully account for the operational characteristics of the metro system when forecasting its outbound passenger flow.

Bus arrival time is affected by various factors like traffic congestion, signal timing, etc. However, the metro is unaffected by these factors and the trains run on more precise timetables. But there are still some problems, as listed below;

1. Metro network is a large and complex system. The operators continuously optimize the operation diagrams of some lines, some sections, and some periods of time; such as reducing headway to increase the number of trains.

2. The number of trains is increased temporarily during the holidays.

3. Even if we have the precise arrival time for train $j$, the passenger walking speed in different age group vary as well as the congestion at the platforms vary in different times of day, so the outbound passenger flow peaks brought by train $j$ will arise later than the train's arrival time. All these aspects depict that the corresponding lag time cannot be accurately calculated. For example, as shown in Fig. 9, the arrival time of train $j$ is $t2$ while the corresponding outbound passenger flow peak arises at $t1$; the lag time $t1 - t2$, however, cannot be calculated accurately.

Therefore, we should consider the time of the outbound passenger flow peaks brought by the train's arrival rather than the

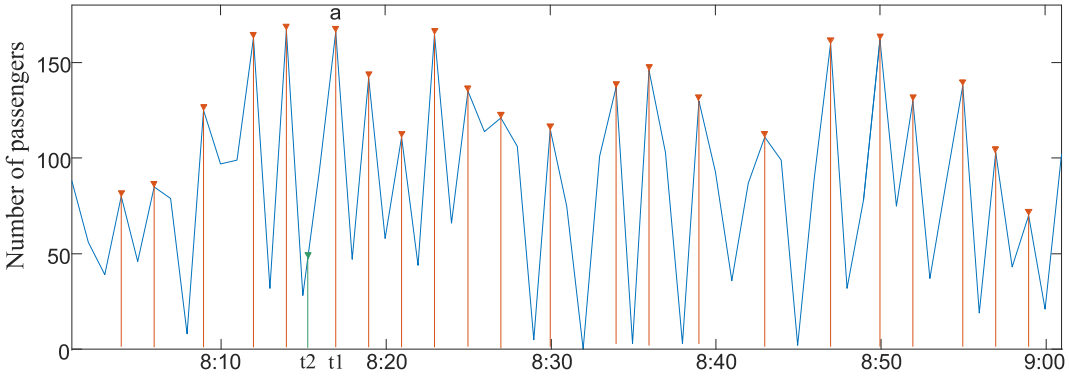


**Fig. 9.** Outbound passenger flow (1-minute interval).

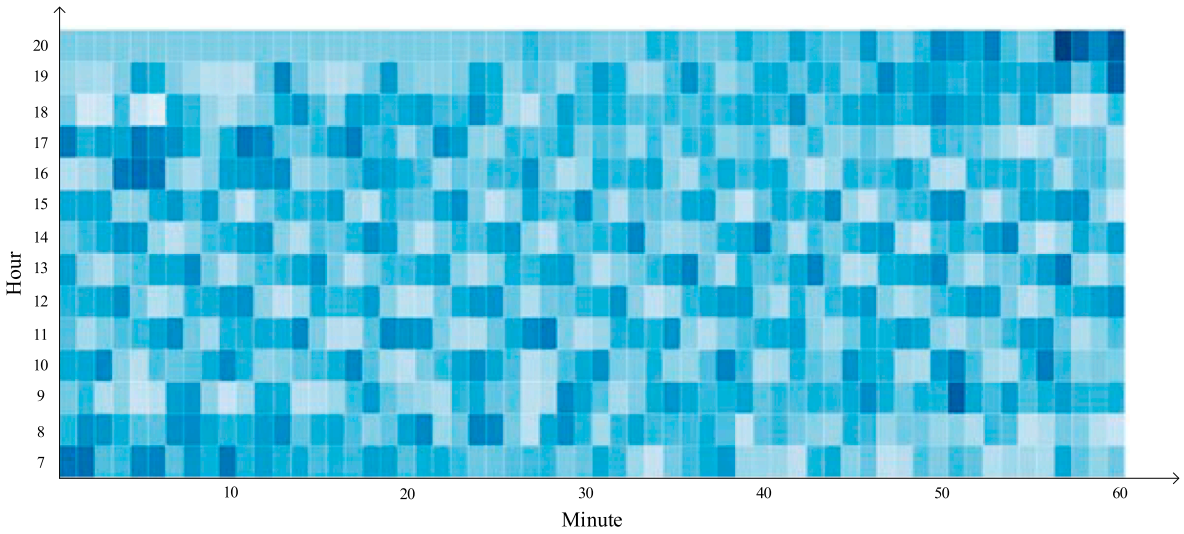


**Fig. 10.** Heat plot for outbound passenger flow (1-minute interval, 7:00–20:00).

timetable itself. So, instead of using the approximate train arrival time, we calculate all the local maximum values (peak values) of the outbound passenger flow according to their peak time. It has two advantages: First, there's no need to acquire every latest detailed timetable and adjustment. As it's completely data-driven; the latest timetable can be learned from the data itself (which solves problem 1 and 2). Second, the train arrival time that depicts the time interval of the local maximum value of passenger flow, passengers' behaviors, and the lag time caused by congestion is taken into consideration (which solves problem 3).

We use $M_t$ to represent passenger flow peaks at $t$-th time interval, where $m_p = 1$ denote there is a passenger flow peak at $p$-th minute in the $t$-th time interval. $m_q^p = 0$ indicates no peak at the corresponding moment.

$$M_t = [m_t^1, m_t^2, ...,m_t^p] \tag{18}$$

### 4.5. Fusion

In this section, we describe how to combine the above components to formulate the overall framework. For different application scenarios, we can choose a different combination of modules. Then, the selected components are fused through a concatenate layer and are appended with a fully-connected layer followed by a single neuron output layer. The concatenate layer takes the different module outputs as the input and then concatenates them into a single vector. The single neuron finally outputs the predicted passenger flow, as shown in Fig. 11.

### 4.6. Training algorithm

We first construct the training samples from the original data. Then, the model is trained via backpropagation and Adaptive Moment Estimation (Adam) method (Kingma and Ba, 2014). The training process is summarized as follows:

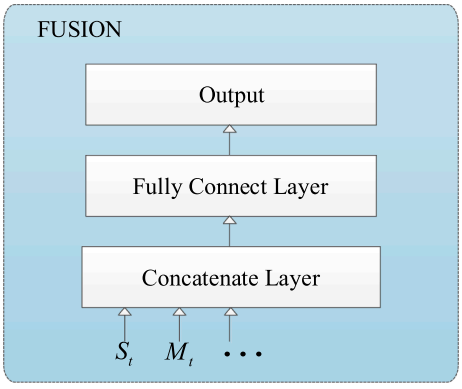


**Fig. 11.** Modules fusion.

Training algorithm

*//Obtain the training set*
**for** all available time interval $t$
*//obtain the flow in nearest time from historical observations (Section 4.2)*
$N_t = [x_{t-1}, x_{t-2}, ...,x_{t-c}]$
*//daily periodicity*
$D_t = [x_{t-d}, x_{t-2d}, ...,x_{t-nd}]$
*// long-term trend*
$L_t = [x_{t-w}, x_{t-2w}, ...,x_{t-mw}]$
*//capture the spatial features (Section 4.3)*
$S_t = [u_1^{r1}, ...,u_1^{r1-e1}, u_1^{r2}, ...,u_1^{r2-e2}, ...,u_k^{r1}, ...,u_k^{r1-e1}, u_k^{r2}, ...,u_k^{r2-e2}]$
*// extract features from metro operation (Section 4.4)*
$M_t = [m_t^1, m_t^2, ...,m_t^p]$
*//add the external environmental factors $E_t$ (Section 4.1)*
*//$x_t$ is the passenger flow at time interval $t$*
sample $(\{N_t, D_t, L_t, S_t, M_t, E_t\}, x_t)$ makes up the training set
**end for**
*//train our model*
initialize parameters in the model
**repeat**
randomly select a batch of samples from the training set
*// train the model through backpropagation and Adam algorithm*
minimize the mean absolute error objective with the training set
**until** the stopping criteria is reached

## 5. Experiments and results

The data collected from the Nanjing Metro System is used as a case study to verify the proposed deep learning-based architecture. The data consists of weekday records from March 18th to April 30th and August 1st to November 9th, 2016. The metro operates from 6:00 am to 22:00 pm. Abnormal data was removed before constructing the final dataset, containing records of 103 days (29664 samples). We chose the last 33 workdays as the testing set while the remaining samples as the training set. The time interval selected for aggregating the passenger flow for prediction in the case study is 10 min. It should be noted that if the selected time interval is too small, the prediction lacks accuracy as well as significance. Moreover, the flows at a short time interval are often trivial (even be zero), which is quite problematic for the prediction method. We use the Min-Max normalization method to scale the passenger flow data in the range $[-1,1]$ for both the training and testing set. During the evaluation, the normalized forecast values are re-scaled and compared with the actual observations.

Performance of the proposed model is compared with three baseline models, presented as follows;

History Data: We use the nearest historical observations as a baseline, e.g. when we predict passenger flow from Monday to Friday, the passenger flow at the previous week is set as the prediction results.

ARIMA: ARIMA is a class of statistical models that are popularly used for time series forecasting. A rolling forecast ARIMA model is adopted for the performance comparison. The proper selection of the parameters; p (AR term), d (The order of difference), and q (MA term) is critical for the ARIMA model. The p, d, and q ARIMA parameters selected in the study are 7,1, and 1 respectively. The optimal parameters can be obtained through grid search.

FNN: Feedforward neural network can capture the complex nonlinear relationship among different variables.

The performance metrics used in the study are Symmetric Mean Absolute Percent Error (SMAPE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Relative Error (MRE).

$$SMAPE = \frac{2}{N}\sum_{i=1}^{N}\frac{|y_i - \hat{y}_i|}{y_i + \hat{y}_i} \times 100\% \tag{19}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{20}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{21}$$

$$MRE = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i - \hat{y}_i|}{y_i} \times 100\% \tag{22}$$

where $y_i$ is the observed passenger flow and $\hat{y}_i$ is the predicted flow during $i$th time interval. $N$ is the total number of predicted passenger flow.

As aforementioned, some modules are already provided with the flexibility for different application scenarios. However, these

modules are to some extent interdependent. The model proposed in the study has multiple variants, including DL-N, DL-NC, DL-NCT, etc., which utilize different components of our architecture.

DL-N indicates the model is **D**eep **L**earning-based and uses the temporal characteristic that passenger flow in nearest time interval shares a higher similarity than the distant ones (i.e. **N**earest part in the architecture). DL-N is the foundation for the prediction task as instead of the individual application, to incorporate temporal features other modules also depend on DL-N. For example, DL-N**C** denotes the daily **C**ycle is incorporated into the model while long-term **T**rend is also considered in the DL-NC**T** model. Likewise, DL-NCT**E** denotes external environmental factors are added to the model.

DL-N**S** denotes the extension of DL-N model that further captures the **S**patial characteristics.

DL-N**M** indicates DL-N model that considers the characteristics of **M**etro operation while forecasting the outbound passenger flow.

A modular framework is further proposed with three basic module settings, namely Module DL-N, DL-C, and DL-T. All the modules contain three layers of LSTM, the node number of which are 32, 32 and 16 respectively. Module DL-S, DL-M, and FNN-N contain three layers of Fully Connected Layer, the node number of which are 32, 32 and 16 respectively. Module DL-E uses an Embedding Layer and a Flatten Layer. The outputs of basic modules are first combined (e.g., the combination of DL-N and other modules). Finally, two Fully Connected Layers with the node number of 16 and 1 are added.

At present, the most popular nonlinear activation function is the rectified linear unit (Lecun et al., 2015). In this study, we used a rectified linear unit in the Fully Connected Layer while sigmoid activation function is used for the gates in LSTM layer. The cell input and output activation functions are tanh. Additionally, as the flow prediction is a regression problem, a linear activation function is used in the last layer. To monitor the internal status and statistics of the network, callback functions, a set of functions to be applied at given stages of the training procedure, are defined. EarlyStopping is set to terminate training when the objective function stops improving. Further, various statistics of the validation set can be displayed at the end of each epoch through a user-defined callback function.

To account for the temporal dependencies, we use the passenger flow in recent four adjacent time intervals (i.e.$[x_{t-1}, x_{t-2}, ...,x_{t-4}]$), the passenger flow at the same time interval in the previous four days (i.e. $[x_{t-d}, x_{t-2d}, ...,x_{t-4d}]$), and the passenger flow at the same time interval in the previous three weeks (i.e. $[x_{t-w}, x_{t-2w}, x_{t-3w}]$). For the spatial features, we sort the daily OD data and select 5 origin stations with the highest passenger flow. As the flow at such stations is high and stable throughout the metro operating period each day, we only consider the data of four stations in this study. Then, the inbound passenger flow data at the $r_1$-th, and $r_2$-th time intervals of the selected stations ($r_1$, $r_2$ can be calculated by Eq. (16) and Eq. (17) respectively) are acquired accordingly. We eliminate the impact of stochastic factors and calculate the average outbound passenger flow for the span of 5 weekdays. Subsequently, the local maximum values are obtained. The minimum interval between two peaks (i.e. the local maximum values) is 2 min while the peak value should be more than 10 people per minute in case of stations with high passenger flow. However, for stations with low passenger flow, this threshold is set at 5 people per minute. Regarding the metadata (i.e. the day of the week, the time of day, and the grade of precipitation), an embedding technique is used to map the categorical values to 3-dimensional vectors. It should be noted that the training set and test set are processed using the same parameters. Then the model itself can learn information that helps improve the accuracy of predictions.

In this study, 3 representative stations are chosen, namely; 1. transfer station, 2. regular station, and 3. regular station with low passenger flows. The proposed deep learning-based prediction architecture is evaluated using SMAPE, RMSE, MAE, and MRE. The results are presented in Tables 1–4.

In terms of SMAPE, RMSE, MAE, and MRE, different variant models were tested and compared with the baseline result to determine the prediction accuracy of the models. From Table 1 and Table 2, we can observe that taking only the passenger flow in nearest time interval into consideration leads to lower prediction as the SMAPE of DL-N model for the transfer station, regular station 1, and regular station 2 is 16.68%, 18.07%, and 26.83% respectively. The RMSE of DL-N model for the transfer station, regular station 1, and regular station 2 are 65.38, 37.90, and 10.64 respectively. Similar results can be observed regarding the MAE and MRE of DL-N model for the three stations. The accuracy remarkably improved when the daily cyclicality characteristic was incorporated in the model, as the SMAPE of DL-NC model for the transfer station, regular station 1, and regular station 2 are 14.48%, 16.34%, and 24.78% respectively, which is better as compared to DL-N model.

**Table 1**
Performance of SMAPE and RMSE in the Validation Set (outbound passenger flow).

| Model | Transfer station | | Regular station 1 | | Regular station 2 | |
|---|---|---|---|---|---|---|
| | SMAPE | RMSE | SMAPE | RMSE | SMAPE | RMSE |
| History Data | 32.40% | 207.32 | 35.71% | 127.23 | 53.20% | 29.07 |
| ARIMA | 25.21% | 96.19 | 22.84% | 55.53 | 41.63% | 11.54 |
| FNN-N | 16.75% | 66.40 | 18.51% | 38.74 | 27.31% | 10.76 |
| DL-N | 16.68% | 65.38 | 18.07% | 37.90 | 26.83% | 10.64 |
| DL-NC | 14.48% | 60.11 | 16.34% | 32.75 | 24.78% | 9.76 |
| DL-NCT | 13.61% | 51.49 | 14.81% | 30.03 | 22.42% | 9.04 |
| DL-NCTE | 12.91% | 50.15 | 14.59% | 29.70 | 22.31% | 8.47 |
| DL-NS | 16.58% | 61.65 | 16.91% | 34.53 | 25.43% | 9.49 |
| DL-NM | 16.45% | 60.36 | 17.50% | 35.69 | 26.07% | 10.00 |
| Best | 12.85% | 48.31 | 14.49% | 28.29 | 22.12% | 8.19 |

**Table 2**
Performance of MAE and MRE in the Validation Set (outbound passenger flow).

| Model | Transfer Station | | Regular Station 1 | | Regular Station 2 | |
|---|---|---|---|---|---|---|
| | MAE | MRE | MAE | MRE | MAE | MRE |
| History Data | 96.88 | 42.52% | 55.85 | 83.23% | 13.58 | 77.03% |
| ARIMA | 57.77 | 35.12% | 30.98 | 29.41% | 7.61 | 52.12% |
| FNN-N | 47.15 | 17.56% | 26.21 | 20.32% | 7.90 | 28.33% |
| DL-N | 45.61 | 17.23% | 25.14 | 19.90% | 7.41 | 26.33% |
| DL-NC | 38.46 | 15.56% | 21.84 | 17.43% | 6.62 | 24.52% |
| DL-NCT | 33.07 | 14.28% | 19.72 | 16.87% | 6.02 | 23.46% |
| DL-NCTE | 32.75 | 13.42% | 18.59 | 15.86% | 5.92 | 22.08% |
| DL-NS | 43.30 | 17.09% | 22.48 | 19.02% | 6.70 | 22.95% |
| DL-NM | 42.02 | 16.64% | 22.95 | 18.99% | 7.06 | 22.93% |
| Best | 31.19 | 12.73% | 18.06 | 15.24% | 5.59 | 21.71% |

**Table 3**
Performance of SMAPE and RMSE in the Validation Set (inbound passenger flow).

| Model | Transfer station | | Regular station 1 | | Regular station 2 | |
|---|---|---|---|---|---|---|
| | SMAPE | RMSE | SMAPE | RMSE | SMAPE | RMSE |
| History Data | 28.61% | 139.25 | 35.01% | 96.72 | 54.05% | 31.44 |
| ARIMA | 19.30% | 75.72 | 20.22% | 42.91 | 39.91% | 12.65 |
| FNN-N | 16.97% | 68.00 | 18.52% | 39.27 | 27.17% | 10.88 |
| DL-N | 16.91% | 65.48 | 17.87% | 37.47 | 27.03% | 10.73 |
| DL-NC | 14.52% | 59.31 | 16.09% | 33.79 | 25.22% | 9.99 |
| DL-NCT | 12.90% | 48.62 | 14.68% | 29.86 | 23.14% | 8.93 |
| DL-NCTE | 12.83% | 48.04 | 14.61% | 29.71 | 22.45% | 8.57 |

**Table 4**
Performance of MAE and MRE in the Validation Set (inbound passenger flow).

| Model | Transfer station | | Regular station 1 | | Regular station 2 | |
|---|---|---|---|---|---|---|
| | MAE | MRE | MAE | MRE | MAE | MRE |
| History Data | 77.78 | 40.01% | 49.56 | 56.05% | 12.12 | 88.05% |
| ARIMA | 55.60 | 28.43% | 28.95 | 24.41% | 9.21 | 47.47% |
| FNN-N | 45.26 | 17.59% | 25.97 | 22.91% | 8.46 | 39.02% |
| DL-N | 44.32 | 17.21% | 23.11 | 20.31% | 7.46 | 32.21% |
| DL-NC | 39.07 | 15.13% | 21.67 | 17.25% | 6.68 | 27.13% |
| DL-NCT | 33.68 | 13.89% | 20.54 | 16.05% | 6.11 | 26.87% |
| DL-NCTE | 33.16 | 13.71% | 18.97 | 15.81% | 5.81 | 25.31% |

The SMAPE was further improved when the weekly trend characteristic was incorporated in the model i.e. from 14.48%, 16.34%, and 24.78% to 13.61%, 14.81%, and 22.42% respectively. The RMSE, MAE, and MRE were also found to improve. However, it should be noted that all the traffic time series data do not have fixed patterns or cycles (e.g., a cycle of 1 day). As metro users are dominated by commuting passengers, the metro passenger flow data reflects the passengers' fixed behavior patterns reasonably well. For example, the variations in today's passenger flow during a certain period of time is similar to that of the previous day. This is the main reason for the improvement in the result while considering time cycle characteristics and time trend characteristics. Likewise, the supply and demand for taxis in urban areas and the traffic flow on urban roads have the same characteristics (i.e., cycle and trend).

The traffic flow is considered a nonlinear system as it frequently changes among free flow, congestion, breakdown and resumed situations. In order to forecast sudden fluctuations in passenger flow effectively, this paper captures these variations from the perspective of passenger flow conservation. From Table 1, the SMAPE of DL-NS model for the transfer station, regular station 1, and regular station 2 is found to reduce from 16.68%, 18.07%, and 26.83% in DL-N to 16.58%, 16.91%, and 25.43% respectively. Likewise, the RMSE, MAE, and MRE of DL-NS model for the transfer station, regular station 1, and regular station 2 also show significant reduction compared to DL-N. Moreover, as more such unusual cases occur, the better will be the result of the inclusion of such characteristics.

Metro operation has various distinctive characteristics, for example, the trains run on more precise timetables. In this paper, we proposed a completely data-driven method that takes the arrival time, passengers' behaviors, and the lag time caused by congestion into consideration. Compared to the SMAPE, RMSE, MAE, and MRE of DL-N model, the performance metrics of DL-NM model are all found to reduce significantly. For example, the SMAPE of DL-NM model for the three stations reduces from 16.68%, 18.07%, and

26.83% in DL-N to 16.45%, 17.50%, and 26.07% respectively.

Based on the real-time evaluation of networks' performance, metro operators continuously optimize the operation diagrams of some lines, some sections, and some periods of time. When the train's schedule changes, it will yield better results with the inclusion of this feature. Moreover, with the consideration of identity features like the day of the week or the time of day, the SMAPE is found to reduce to 12.91%, 14.59%, and 22.31% respectively. The other three metrics also reduces to some extent. It is worth noting that the best result shown in the above table is achieved by the combination model considering all the possible features. The performance metric values of the best performing model are 12.85%, 14.49%, and 22.12% (SMAPE); 48.31, 28.29, and 8.19 (RMSE); 31.19, 18.06, and 5.59 (MAE); 12.73%, 15.24%, and 21.71% (MRE) respectively. These results depict that various components should be considered for optimized prediction results.

For the inbound passenger flow prediction, there is only minor useful information available compared to outbound flow. Like the above outbound flow results (Table 1 and Table 2), the addition of both the daily cycle and trend component into the models improved the prediction performance.

The experiment shows that the proposed model performs well in predicting passenger flow. In the following discussions, we extend the model to general conditions to fit the diverse constraints in the actual situation.

DL-N is the basic model in which the test set has less demand for data volumes. DL-C module is suitable for data with a clear cycle pattern, for example, metro/bus passenger flow, taxi supply and demand, urban traffic flow, etc. Due to the existence of a large number of commuter passengers, such data show significant cyclicality. However, the test set requires only a few days' data otherwise the module cannot be used. For instance, in the actual forecast, may be due to business needs, we may only use the previous one hour's data to predict for the desired time interval. DL-T is suitable for data with an upward or downward trend. Similarly, there is a requirement of several weeks' time span in the test data. DL-E can improve the prediction accuracy in cases where the data distribution does not change. There are many reasons for the change in the distribution of data, for example, if the first vehicle operation time of the metro line postponed by 30 min, the peak of the corresponding time would also change. If the first vehicle time is different in the training set and the test set, it is evident that the distribution of data will be different. In such a case, incorporating the day of the week and the time of day in the model will not work.

The above four modules are suitable for inbound and outbound passenger flow. If the flow data lacks cycle and trend, the data time span in the test set is too short, or the data distribution changes, then the above components which improve the accuracy of the model cannot be used whereas using the basic module only may be ineffective. In such cases, while forecasting outbound passenger flow, we can consider trying DL-S and DL-M component to improve prediction. In the future study, we plan to incorporate the corresponding module for the inbound passenger flow to deal with the above situation. As DL-S module does not have the requirement of data time span but requires data for the entire network, the module can effectively predict the sudden rise or fall of passenger flow. Furthermore, the train schedule changes frequently, making it difficult to obtain the precise updated schedule data. DL-M is a data-driven module based on the raw fine-grained passenger traffic flow data which is a more practical and reliable approach.

## 6. Conclusions

This paper proposed a deep learning-based architecture integrating the domain knowledge in transportation modeling. First, the theoretical aspects of the factors were analyzed for the superior performance of recurrent neural network over feedforward neural network in traffic time series forecasting. Then, the influence of three temporal properties was analyzed to promote general application scenarios. The metro network has unique topological structure. Thus, we used the average travel time instead of the geographical distance and further constructed the features through inflow and outflow conservation to capture the spatial characteristics between stations. Due to the significance of urban rail timetable, we proposed a completely data-driven method taking the train arrival time, passengers' behaviors, and the lag time caused by congestion into consideration.

For the flow prediction problems, there are many similarities between classic transport models and feature engineering in machine learning. Therefore, the domain knowledge from existing transport models can be used to design advanced features. The rationale of this idea is underpinned by the experiments. The results show that the accuracy can be improved when the daily cyclicality characteristic and the weekly trend characteristic are incorporated in the model, indicating that commuters have fixed behavior patterns. As a data-driven method, the endogenous characteristics of the data are further investigated. Compared to the base model, all the metrics decrease when the customized design features for schedule are taken into consideration. Based on the flow conservation principle that the outflow always counterbalances the inflow, high-level features are designed, and a reduction in error can be observed in comparison to the base model. This controlled experiment verifies that the domain knowledge can be combined with machine learning to obtain significant input features.

This research is an initial step in exploring short-term metro passenger flow prediction using a new generation technology. As a future work, it is worthwhile to further investigate the essential causes for the effectiveness of a particular feature and to enable machine learning better comprehend the characteristics, instead of taking algorithm entirely as a black box. To better utilize multi-source data and investigate causality between different data is necessary. In the future study, it is needed to combine the transportation and digital signal processing theory from the perspectives of time domain and frequency domain, to eliminate its impact on the results.

## Acknowledgment

This study is supported by the General Projects (No. 71771050) and Key Projects (No. 51638004) of the National Natural Science Foundation of China, and the Postgraduate Research&Practice Innovation Program of Jiangsu Province (KYCX18_0150).

## References

Ahmed, M.S., Cook, A.R., 1979. Analysis of freeway traffic time-series data by using Box-Jenkins techniques. Transp. Res. Record: J. Transp. Res. Board 722, 1–9.
Bengio, Y., 2009. Learning Deep Architectures for AI. *Foundations and Trends®*. Mach. Learn. 2 (1), 1–127. https://doi.org/10.1561/2200000006.
Bengio, Y., Schwenk, H., Senécal, J.-S., Morin, F., Gauvain, J.-L., 2006. Neural Probabilistic Language Models. In: In: Holmes, D.E., Jain, L.C. (Eds.), Innovations in Machine Learning, vol. 94. Springer-Verlag, Berlin/Heidelberg, pp. 137–186. https://doi.org/10.1007/3-540-33486-6_6.
Bie, Y., Gong, X., Liu, Z., 2015. Time of day intervals partition for bus schedule using GPS data. Transp. Res. Part C: Emerg. Technol. 60, 443–456. https://doi.org/10.1016/j.trc.2015.09.016.
Chan, K.Y., Dillon, T.S., Singh, J., Chang, E., 2012. Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and Levenberg–Marquardt algorithm. IEEE Trans. Intell. Transp. Syst. 13 (2), 644–654. https://doi.org/10.1109/TITS.2011.2174051.
Chang, H., Lee, Y., Yoon, B., Baek, S., 2012. Dynamic near-term traffic flow prediction: system-oriented approach based on past experiences. IET Intel. Transport Syst. 6 (3), 292–305. https://doi.org/10.1049/iet-its.2011.0123.
Collobert, R., Weston, J., 2008. In: A unified architecture for natural language processing: deep neural networks with multitask learning. ACM Press, Helsinki, Finland, pp. 160–167. https://doi.org/10.1145/1390156.1390177.
Duan, Y., Lv, Y., Liu, Y.-L., Wang, F.-Y., 2016a. An efficient realization of deep learning for traffic data imputation. Transp. Res. Part C: Emerg. Technol. 72, 168–181. https://doi.org/10.1016/j.trc.2016.09.015.
Duan, Y., Lv, Y., Wang, F.-Y., 2016b. In: Performance evaluation of the deep learning approach for traffic flow prediction at different times. IEEE, Beijing, China, pp. 223–227.
Duan, Y., Lv, Y., Wang, F.-Y., 2016bc. Travel time prediction with LSTM neural network. In: Presented at the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). IEEE, Rio de Janeiro, Brazil, pp. 1053–1058.
Gang, X., Kang, W., Wang, F., Zhu, F., Lv, Y., Dong, X., Pirttikangas, S., 2015. Continuous travel time prediction for transit signal priority based on a deep network. Presented at the 2015 IEEE 18th International Conference on Intelligent Transportation Systems 523 528 Gran Canaria, Spain. IEEE.
Graves, A., Mohamed, A., Hinton, G., 2013. In: Speech recognition with deep recurrent neural networks. IEEE, Vancouver, BC, Canada, pp. 6645–6649. https://doi.org/10.1109/ICASSP.2013.6638947.
Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y., 2016. Deep learning. MIT press, Cambridge.
Gu, Z., Saberi, M., Sarvi, M., Liu, Z., 2018. A big data approach for clustering and calibration of link fundamental diagrams for large-scale network simulation applications. Transp. Res. Part C: Emerg. Technol. 94, 151–171. https://doi.org/10.1016/j.trpro.2017.05.050.
Hansen, J.V., McDonald, J.B., Nelson, R.D., 1999. Time series prediction with genetic-algorithm designed neural networks: an empirical comparison with modern statistical models. Comput. Intell. 15 (3), 171–184. https://doi.org/10.1111/0824-7935.00090.
Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Kingsbury, B., 2012. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process Mag. 29 (6), 82–97. https://doi.org/10.1109/MSP.2012.2205597.
Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.
Huang, W., Song, G., Hong, H., Xie, K., 2014. Deep architecture for traffic flow prediction: deep belief networks with multitask learning. IEEE Trans. Intell. Transp. Syst. 15 (5), 2191–2201. https://doi.org/10.1109/TITS.2014.2311123.
Huang, G.B., Zhu, Q.Y., Siew, C.K., 2006. Extreme learning machine: theory and applications. Neurocomputing 70 (1), 489–501. https://doi.org/10.1016/j.neucom.2005.12.126.
Jiao, P., Li, R., Sun, T., Hou, Z., Ibrahim, A., 2016. Three revised kalman filtering models for short-term rail transit passenger flow prediction. Math Probl Eng 2016, 1–10. https://doi.org/10.1155/2016/9717582.
Jeong, Y.-S., Byon, Y.-J., Castro-Neto, M.M., Easa, S.M., 2013. Supervised weighting-online learning algorithm for short-term traffic flow prediction. IEEE Trans. Intell. Transp. Syst. 14 (4), 1700–1707. https://doi.org/10.1109/TITS.2013.2267735.
Kingma, D.P., Ba, J., 2014. Adam: A Method for Stochastic Optimization. ArXiv:1412.6980 [Cs]. Retrieved from http://arxiv.org/abs/1412.6980.
Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Presented at the Advances in neural information processing systems, Lake Tahoe, United States, pp. 1097–1105.
LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444. https://doi.org/10.1038/nature14539.
Liu, Y., Jia, R., Xie, X., Liu, Z., 2018. A two-stage destination prediction framework of shared bicycle based on geographical position recommendation. IEEE Intell. Transp. Syst. Mag. 1–1. https://doi.org/10.1109/MITS.2018.2884517.
Liu, Z., Wang, S., Chen, W., Zheng, Y., 2016. Willingness to board: A novel concept for modeling queuing up passengers. Transp. Res. Part B: Methodol. 90, 70–82. https://doi.org/10.1016/j.trb.2016.04.005.
Liu, Z., Yan, Y., Qu, X., Zhang, Y., 2013. Bus stop-skipping scheme with random travel time. Transp. Res. Part C: Emerg. Technol. 35, 46–56. https://doi.org/10.1016/j.trc.2013.06.004.
Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F.-Y., 2014. Traffic flow prediction with big data: a deep learning approach. IEEE Trans. Intell. Transp. Syst. 16 (2), 865–873. https://doi.org/10.1109/TITS.2014.2345663.
Ma, X., Tao, Z., Wang, Y., Yu, H., Wang, Y., 2015. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transp. Res. Part C: Emerg. Technol. 54, 187–197. https://doi.org/10.1016/j.trc.2015.03.014.
Miles, J.C., Walker, A.J., 2006. The potential application of artificial intelligence in transport. IEE Proc. – Intell. Transport Syst. 153 (3), 183–198. https://doi.org/10.1049/ip-its:20060014.
Niu, H., Zhou, X., 2013. Optimizing urban rail timetable under time-dependent demand and oversaturated conditions. Transp. Res. Part C: Emerg. Technol. 36, 212–230. https://doi.org/10.1016/j.trc.2013.08.016.
Pereira, F.C., Rodrigues, F., Ben-Akiva, M., 2015. Using data from the web to predict public transport arrivals under special events scenarios. J. Intell. Transp. Syst. 19 (3), 273–288. https://doi.org/10.1080/15472450.2013.868284.
Polson, N.G., Sokolov, V.O., 2017. Deep learning for short-term traffic flow prediction. Transp. Res. Part C: Emerg. Technol. 79, 1–17. https://doi.org/10.1016/j.trc.2017.02.024.
Sun, S., Zhang, C., Yu, G., 2006. A Bayesian network approach to traffic flow forecasting. IEEE Trans. Intell. Transp. Syst. 7 (1), 124–132. https://doi.org/10.1109/TITS.2006.869623.
Sun, Shiliang, Huang, R., Gao, Y., 2012. Network-scale traffic modeling and forecasting with graphical lasso and neural networks. J. Transp. Eng. 138 (11), 1358–1367. https://doi.org/10.1061/(ASCE)TE.1943-5436.0000435.
Tang, K., Chen, S., Liu, Z., 2018. A tensor-based bayesian probabilistic model for citywide travel time estimation using sparse trajectories. Transp. Res. Part C: Emerg. Technol. 90, 260–280. https://doi.org/10.1016/j.trc.2018.03.004.
Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C., 2005. Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach. Transp. Res. Part C: Emerg. Technol. 13 (3), 211–234. https://doi.org/10.1016/j.trc.2005.04.007.
Wei, Y., Chen, M.-C., 2012. Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks. Transp. Res. Part C: Emerg. Technol. 21 (1), 148–162. https://doi.org/10.1016/j.trc.2011.06.009.

Williams, B., 2001. Multivariate vehicular traffic flow prediction: evaluation of ARIMAX modeling. Transp. Res. Record: J. Transp. Res. Board 1776, 194–200. https://doi.org/10.3141/1776-25.

Williams, B., Durvasula, P., Brown, D., 1998. Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models. Transp. Res. Record: J. Transp. Res. Board 1644, 132–141. https://doi.org/10.3141/1644-14.

Williams, B.M., Hoel, L.A., 2003. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results. J. Transp. Eng. 129 (6), 664–672. https://doi.org/10.1061/(ASCE)0733-947X(2003) 129:6(664).

Wu, J., Liu, M., Sun, H., Li, T., Gao, Z., Wang, D.Z.W., 2015. Equity-based timetable synchronization optimization in urban subway network. Transp. Res. Part C: Emerg. Technol. 51, 1–18. https://doi.org/10.1016/j.trc.2014.11.001.

Wu, Y., Tan, H., Qin, L., Ran, B., Jiang, Z., 2018. A hybrid deep learning based traffic flow prediction method and its understanding. Transp. Res. Part C: Emerg. Technol. 90, 166–180. https://doi.org/10.1016/j.trc.2018.03.001.

Zhang, G., Eddy Patuwo, B., Hu, Y.M., 1998. Forecasting with artificial neural networks. Int. J. Forecast. 14 (1), 35–62.

Zhang, J., Zheng, Y., Qi, D., 2017. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction (p. 7). Presented at the AAAI Conference on Artificial Intelligence. AAAI, San Francisco, California, USA.

Zhong, C., Batty, M., Manley, E., Wang, J., Wang, Z., Chen, F., Schmitt, G., 2016. Variability in regularity: mining temporal mobility patterns in London, singapore and beijing using smart-card data. PLoS One 11 (2), e0149222. https://doi.org/10.1371/journal.pone.0149222.